生成型一文要約のためのマルチアテンションモデルの提案

吉岡 重紀† 山名 早人‡

†早稲田大学 基幹理工学研究科 〒169-8050 東京都新宿区大久保 3-4-1
‡早稲田大学 理工学術院 〒169-8050 東京都新宿区大久保 3-4-1
国立情報学研究所 〒101-8430 東京都千代田区一ツ橋 2-1-2

E-mail: †‡{s.yoshioka, yamana}@yama.info.waseda.ac.jp

あらまし 世の中のデータは指数関数的に増加しており，その量は 2020 年には 44ZB に達すると予想されている．今後，膨大なデータ全てを読むことはますます難しくなり， データの取捨選択や圧縮が必須となる．自動要約は文書を要旨に圧縮しており，元文書を読むべきかの判断や，短い文書から内容理解をすることが可能となる．自動要約は抽出型要約と生成型要約の二種類に大別できる．抽出型要約は元文書に含まれる文やフレーズを組み合わせることで要約を作る．一方，生成型 要約は元文書の内容を元に，新たな文を生成して要約を行う．そのため，生成型要約では，人間が行うような言い換えや一般化，並び替えによる要約が可能となる．しかし，既存研究の多くは抽出型要約によるもので，生成型要約の研究は文から短い文を生成する文レベルの要約にとどまっている．そこで本研究では，アテンションモデルを複数用いることにより複数文の文書から一文の要約を生成型要約で行う方法を提案する．嗜好テストによる実験を行い，既存手法より 90%信頼区間で元文書に対し適切な要約を生成することができた．

キーワード 生成型要約，アテンションモデル

1. はじめに

現在，世の中のデータは指数関数的に増加しており，その量は 2020 年には 44ZB に達すると予想されている[1]．今後，データはより膨大な量となり，人間が全てのデータを直接利用することはますます難しくなる．そのため，データの取捨選択や圧縮が必須となる．文書データにおける取捨選択や圧縮の手段として要約がある．要約は文書を要旨に圧縮するため，元文書を読むべきかの判断や要約だけで元文書の内容理解が可能となる．

要約をプログラムにより作成する自動要約は，要約の作り方により抽出型要約と生成型要約の二種類に大別できる．抽出型要約は元文書に含まれる文やフレーズを組み合わせることで要約を作る．一方，生成型要約は元文書の内容をもとに，新たな文を生成して要約を行う．人間は言い換えや一般化，並び替えなどを行い，元文書とは異なる表現を使いながら要約を行う[1]．抽出型要約では元文書に使われている表現しか用いることができず,人間の要約に近づけるのに限界がある．一方，生成型要約では元文書に含まれる単語に制限されないため，元文書と異なる表現で要約を作ることも可能となる．したがって，生成型要約は人間が要約を作る方法に近く，人間が作る要約に近いものが期待できる．しかし，既存研究の多くは抽出型要約によるもので，生成型要約の研究は文から短い文を生成する文レベルの要約にとどまっている．

生成型要約を行うには，まず，元文書の内容をコンピュータの扱える中間表現に直し，次にその中間表現から文生成を行わなくてはならない．しかし，文書の内容を忠実に表現可能な中間表現方法は確立していない．その上，中間表現が可能になっても，その中間表現が表す内容を元文書より短くなるように文生成することは難しい課題である．このように生成型要約は技術的課題が多く今まであまり行われて来なかった．

2014 年に機械翻訳のタスクで RNN（Recurrent Neural Network）や LSTM（Long Short Term Memory）などの再帰型ニューラルネットワークを用いた，Encoder-Decoder モデルが登場した[2][3]．２つの再帰型ニューラルネットワークを用い，文書からのベクトル化，ベクトルからの文書生成を行い，直接文書から別言語の文書へ end-to-end で翻訳することを実現した．文書からベクトル化することとベクトルから文書生成することはそれぞれエンコードとデコードと呼ばれ，ベクトルは文書の内容を表す中間表現となっている．2015 年には単語の出力ごとに過去の出力単語と元文書からエンコードを行うテンションモデルが登場した[4]．アテンションモデルは Encoder にも出力単語を入力することにより，次の単語出力で着目すべき元文書中の単語に重み付けしてエンコードすることができる．Encoder-Decoder モデルの機械翻訳は生成型要約の方

[1] Data Growth, Business Opportunities, and the IT Imperatives, http://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm, (2016 年 1 月 5 日アクセス)

法に非常に近く，アテンションモデルを用いた文レベルの生成型要約が提案された[5].

本研究では，これまで実施されてこなかった複数文から構成される文書から要約を生成することに挑戦する．具体的には，アテンションモデルを複数用いることにより複数文の文書から一文の要約を生成型要約で行うマルチアテンションモデルの提案する．産経ニュース[2]からニュース記事とタイトルを収集し，タイトルをニュース記事の一文要約として実験を行う．

本稿では次の構成をとる．まず，2 節で関連研究について述べ，3 節で提案するマルチアテンションモデルについて説明する．4 節で実験と評価を行い，最後に 5 節でまとめる．

2. 関連研究

生成型要約は，1)元文書からその内容を表す中間表現に直し，次に 2)その中間表現から元文書より短くなるように文書生成という２つの行程から構成される．近年，機械翻訳の分野でニューラルネットワークを用いた Encoder-Decoder モデルが提案された．このモデルは生成型要約の行程と類似しており，既存の文レベルの生成型要約の研究ではニューラルネットワークを用いた機械翻訳を参考にしている．本節では 2.1，2.2 でそれぞれ Encoder-Decoder モデルの機械翻訳とアテンションモデルについて説明し，2.3 でアテンションモデルを利用して文レベルの生成型要約を行った既存研究について紹介する．

2.1. Encoder-Decoder モデルによる機械翻訳

Cho らは二つの RNN を用いて機械翻訳を行う RNN Encoder-Decoder モデルの提案を行った[2]．二つの RNN の一方で，言語 A で記述された文をベクトル化し，もう一方で，そのベクトルから言語 B の文を生成する．ここで，前者は，RNN を文からベクトル表現にしていることからエンコーダ，後者は，RNN をベクトルから文にしていることからデコーダと呼ばれる．図 **1** に Encoder-Decoder モデルを示す．

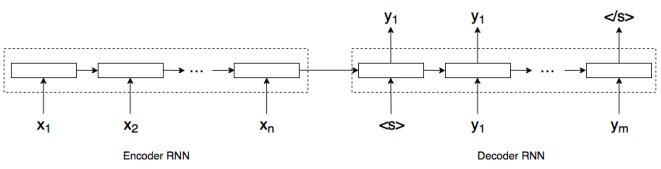

図 1: Encoder-Decoder モデル．$\boldsymbol{x_i}, \boldsymbol{y_i} \in \{\boldsymbol{0},\boldsymbol{1}\}^{\boldsymbol{V}}$，$\boldsymbol{V}$は語彙数．$\boldsymbol{x_i}$および$\boldsymbol{y_i}$は単語を 1-of-K 表現にしたベクトル．（[2][3]を基に作成）

[2] 産経ニュース, http://www.sankei.com/, (2016 年 1 月 5 日アクセス)

Encoder RNN では RNN の入力に 1-of-K 表現された元文書の単語を順に入力し，全ての単語を入力し終わったときの隠れ層の出力を入力文のベクトル表現としている．Decoder RNN は RNNLM(Recurrent Neural Network Language Model)[6]に潜在状態として Encoder RNN から出力されたエンコード結果を入力したものとなっている．学習は交差エントロピー誤差を用い，Decoder から Encoder まで逆伝搬して学習を行っている．このようにすることで end-to-end で機械翻訳をすることを可能とした．

RNN は，原理的には隠れ層は全ての入力を考慮することが可能であるが，実際には長期的な記憶は困難である．そこで，Sutskever らは LSTM を用いたモデルを提案した[3]．RNN よりも LSTM を用いたモデルの方が長期的な記憶が可能となり精度向上をしたが，なおも入力文が長いものは精度が低い傾向となった．これは，入力文は可変長であるのに対し，RNN や LSTM の隠れ層のノード数は定数個であるため，可変長のものから固定長のベクトルにする際に次元数が不足し，情報のロスが起きているためである．

2.2. アテンションモデル

RNN や LSTM を用いた Encoder-Decoder モデルでは，可変長の文を固定長のベクトルにエンコードするため，長い入力文になるほど次元数が不足し，場合精度が下がる問題がある．この問題に対し，Bahdanau らはアテンションモデルを用いた機械翻訳を提案した[4]．図 **2** にアテンションモデルを用いた機械翻訳のモデルを示す．

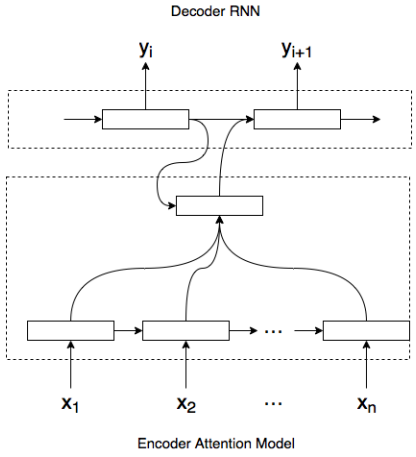

図 2: アテンションモデルを用いた機械翻訳のモデル（[4]を基に作成）

アテンションモデルでは過去の出力から，入力文のどの単語を着目するかの荷重を決定しながら，エンコードを行うモデルである．このようにすることで，長

い文書でも入力文の一部分を着目しながら文生成が可能となり，長期記憶が不要となった．

RNN の Encoder-Decoder モデルとアテンションモデルで比較実験を行い，BLEU の評価はそれぞれ 26.71%と 34.16%となった．また，アテンションモデルは長い入力文でも BLEU の値が下がらないことを示した．

2.3. アテンションモデルを用いた文レベルの生成型要約

Rush らはアテンションモデルを用い，文レベルの生成型要約を提案した[5]．アテンションモデルの入力文の着目すべき単語に荷重を置きエンコードするという特徴から，要約する際に着目すべき箇所を特定しながら，生成的に文レベルの要約を可能とした．以下にアテンションモデルによるエンコーダを示す．

$$encoder(\boldsymbol{x}, \boldsymbol{y}_c) = \boldsymbol{p}^{\mathrm{T}}\bar{\boldsymbol{x}}$$

$$\boldsymbol{p} \propto \exp\left(\tilde{\boldsymbol{x}}\boldsymbol{P}\tilde{\boldsymbol{y}}'_c\right)$$

$$\forall i \ \bar{\boldsymbol{x}}_i = \sum_{q=i-Q}^{i+Q} \frac{\tilde{\boldsymbol{x}}_q}{Q}$$

$$\tilde{\boldsymbol{y}}'_c = (\boldsymbol{G}\boldsymbol{y}_{i-C+1}, \dots, \boldsymbol{G}\boldsymbol{y}_i)$$

$$\tilde{\boldsymbol{x}} = [\boldsymbol{F}\boldsymbol{x}_1, \dots, \boldsymbol{F}\boldsymbol{x}_M]$$

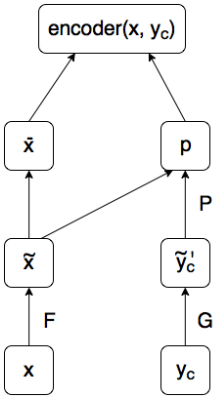

図 3: アテンションモデルによるエンコーダ．（[5]を基に作成）

入力$\boldsymbol{x}$, $\boldsymbol{y}_c$はそれぞれ元文書の単語と過去出力した C 個の単語である．パラメータ$\boldsymbol{F}, \boldsymbol{G}, \boldsymbol{P}$はそれぞれ，$\boldsymbol{F} \in \mathbb{R}^{V \times D}$，$\boldsymbol{G} \in \mathbb{R}^{V \times D}$，$\boldsymbol{P} \in \mathbb{R}^{(CD) \times H}$である．$\boldsymbol{F}, \boldsymbol{G}$はエンベディング行列であり，$V$は語彙数，$D$はエンベディングサイズである．$H$は隠れ層のノード数で，$Q$はスムージングウィンドウ幅である．$\tilde{\boldsymbol{x}}$，$\tilde{\boldsymbol{y}}'_c$は$\boldsymbol{x}$，$\boldsymbol{y}_c$をエンベディングしたベクトルであり，**$\bar{\boldsymbol{x}}$は$\tilde{\boldsymbol{x}}$をスムージングしたベクトルである**．$\boldsymbol{p}$が荷重となっており，$\bar{\boldsymbol{x}}$と掛け合わせることで入力文のうち注目すべき単語に荷重を与えながらエンコーディングをおこなっている．

要約で使うべき単語に着目しながら，生成的に要約を行った．学習にはニュース記事のデータセットである Gigaward データセット[3]を用いた．ヘッドラインを要約として扱い，記事の最初の一文からヘッドラインを生成するように学習を行った．DUC（Document Understanding Conference）[4]が用意している DUC-2004 のデータセットと Gigaward データセットで実験を行い，ROUGE[7]の評価は表 **1** のようになった．

表 1: 既存研究における ROUGE の評価

データセット	ROUGE-1	ROUGE-2	ROUGE-L
DUC-2004	0.2818	0.0849	0.2381
Gigaward	0.3100	0.1265	0.2834

3. マルチアテンションモデル

生成型要約の既存研究では文レベルのものにとどまっている．そこで本研究では，複数のアテンションモデルを用いることで，複数文の文書から一文要約するモデルを提案する．

生成型要約のモデルは図 **4** のように Encoder 部分と Decoder 部分に分けられる．Encoder 部分では元文書をベクトル化し，Decoder 部分では Encoder の出力と過去の出力から次の出力単語を決定する言語モデルとなっている．Encoder 部分で複数のアテンションモデルを用いることで，複数文の文書のエンコードを行い，複数文から一文要約生成を可能とする．

本節では 3.1 に Decoder 部分の言語モデルについて説明し，3.2 で Encoder 部分のマルチアテンションモデルについて説明する．3.3 では学習方法について説明し，3.4 に要約文生成アルゴリズムについて説明する．

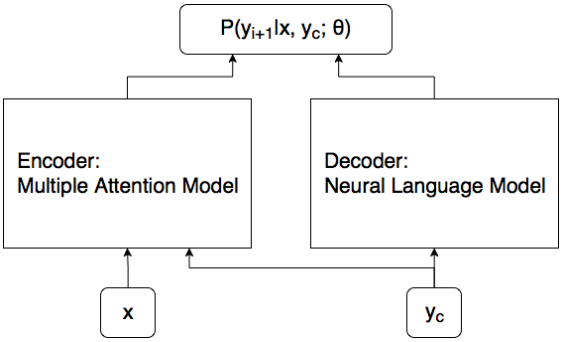

図 4: 生成型要約のモデルの全体像．$\boldsymbol{x}$は元文書の単語，$\boldsymbol{y}$は生成する要約の単語，$\boldsymbol{\theta}$はパラメータを示す．$\boldsymbol{y}_c$は過去 C 個の出力した単語．

[3] English Gigaward, https://catalog.ldc.upenn.edu/LDC2003T05,（2016 年 1 月 8 日アクセス）

[4] Document Understanding Conference, http://duc.nist.gov/,（2016 年 1 月 8 日アクセス）

3.1. 言語モデル

Decoder となる言語モデルには FFNN（Feed Forward Neural Network）による言語モデル[8]を用いた．以下に Decoder のモデルを示す．

$$p(\boldsymbol{y}_{i+1}|\boldsymbol{y}_c, \boldsymbol{x};\, \theta) \propto exp(\boldsymbol{V}\boldsymbol{h} + \boldsymbol{W}encoder(\boldsymbol{x}, \boldsymbol{y}_c))$$
$$\boldsymbol{h} = \tanh\,(\boldsymbol{U}\tilde{\boldsymbol{y}}_c)$$
$$\tilde{\boldsymbol{y}}_c = (\boldsymbol{E}\boldsymbol{y}_{i-C+1}, \ldots, \boldsymbol{E}\boldsymbol{y}_i)$$
$$\boldsymbol{y}_c = [\boldsymbol{y}_{i-C+1}, \ldots, \boldsymbol{y}_i]$$

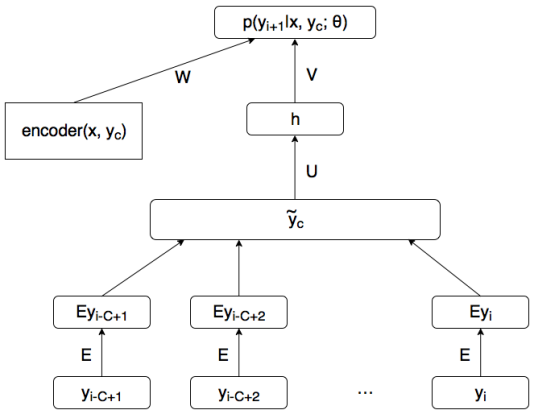

図 5: Decoder の FFNN による言語モデル．

パラメータ θ は $\boldsymbol{E}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$ でそれぞれ，$\boldsymbol{E} \in \mathbb{R}^{V \times D}$，$\boldsymbol{U} \in \mathbb{R}^{(CD) \times H}$，$\boldsymbol{V} \in \mathbb{R}^{H \times H}$，$\boldsymbol{W} \in \mathbb{R}^{H \times H}$ である．$\boldsymbol{E}$ はエンベディング行列であり，V は語彙数，D はエンベディングサイズである．H は隠れ層のノード数である．

3.2. マルチアテンションモデル

複数文に対応するために文から注目単語を抽出するアテンションモデルを用意し，元文書中の各文の注目単語を抽出したうえで，抽出された注目単語に重み付けを行い，文書全体の注目単語を決定する．図 **6** にマルチアテンションモデルを図示する．各文の重み付けの方法として 3.2.1 に平均マルチアテンションモデル，3.2.2 にディープマルチアテンションモデルについて説明する．

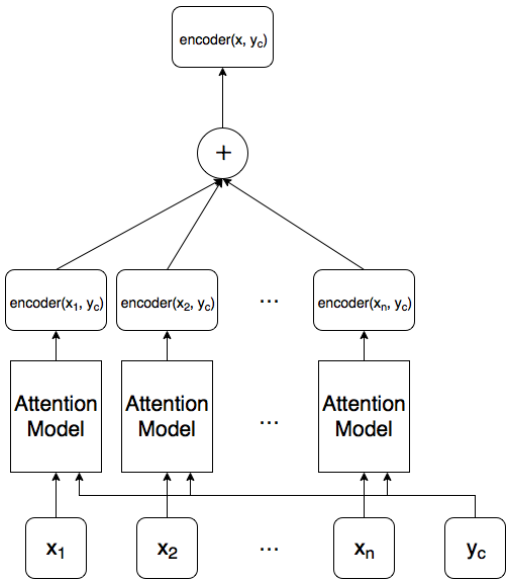

図 6: マルチアテンションモデル．各文のアテンションモデルは図 3 のモデルを用い，全て同じモデルを用いる．$\boldsymbol{x_i} \in \boldsymbol{x}$ で $\boldsymbol{x_i}$ は $\boldsymbol{i}$ 番目の文の単語を表す．

3.2.1. 平均マルチアテンションモデル

各文のアテンションモデルのエンコード結果を平均したものを文書のエンコードとするモデル．各文が同程度反映されたものとなり，実質荷重なしのモデルとなっている．以下にモデル式を示す．

$$encode(\boldsymbol{x}, \boldsymbol{y}_c) = \sum_{i=1}^{n} \frac{encode(\boldsymbol{x}_i, \boldsymbol{y}_c)}{n}$$

3.2.2. ディープマルチアテンションモデル

文をエンコーディングするアテンションモデルの他に各文のエンコード結果をエンコードするアテンションモデルを用意し，文間の荷重を決定して，文書のエンコードを行うモデル．

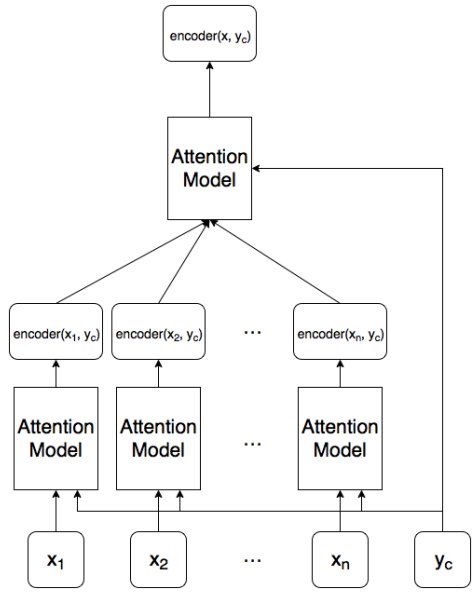

図 7: ディープマルチアテンションモデル．文のアテンションモデルと文間のアテンションモデルは別のモデルを用いる．

3.3. 学習方法

出力は$p(\boldsymbol{y}_{i+1}|\boldsymbol{y}_c, \boldsymbol{x};\theta)$で各単語の確率値が出力される．この単語の確率値の負の対数尤度(Negative Log-Likelihood(NLL))を最小化するように勾配降下法で学習を行う．J個の元文書，参照要約の対$(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}), \ldots, (\boldsymbol{x}^{(J)}, \boldsymbol{y}^{(J)})$があるとき，パラメータ$\theta$の負の対数尤度は以下の式となる．$N$は$\boldsymbol{y}^{(j)}$の長さを表す．

$$NLL(\theta) = -\sum_{j=1}^{J} \log p\left(\boldsymbol{y}^{(j)}|\boldsymbol{x}^{(j)};\theta\right)$$

$$= -\sum_{j=1}^{J}\sum_{i=1}^{N-1} \log p\left(\boldsymbol{y}_{i+1}^{(j)}|\boldsymbol{x}^{(j)}, \boldsymbol{y}_c;\theta\right)$$

学習エポックおよび学習率は次のアルゴリズムのように計画を行った．

アルゴリズム 1 学習計画

Input: 最大エポック数$MaxEpoch$, 初期学習率ϵ, 学習率減衰率d, 最小改善率$MinImp$, パラメータθ

$flag \leftarrow false$
$lastErr \leftarrow inf$
$last\theta \leftarrow \theta$
for $i = 1\ to\ MaxEpoch$ **do**
　トレーニング
　$\theta \leftarrow \theta - \epsilon\nabla E$

　バリデーション
　$err \leftarrow E(\theta)$
　if $lastErr < err$ **do**
　　$\theta \leftarrow last\theta$
　end if
　if $lastErr < err * (1 + MinImp)$ **do**
　　if $flag$ **do**
　　　break
　　else
　　　$\epsilon \leftarrow \epsilon * d$
　　　$flag = true$
　　end if
　else
　　$flag = false$
　end if
end for

3.4. 生成文探索アルゴリズム

要約を生成する際，最も確率が高い文を生成することが好ましいが，最大のものを探索するのは NP 困難な問題である．ビタビアルゴリズムで探索を行う場合，$O(NV^C)$の計算量が必要となる．多くの場合Vは十分に大きく困難である．

本研究では生成文の探索アルゴリズムとしてビームサーチを用いた．

アルゴリズム 2 ビームサーチ

Input: パラメータθ, ビームサイズK, 元文書x
Output: 近似の K-best の 要約
$\pi[0] \leftarrow \{\epsilon\}$
for $i = 0\ to\ N-1$ **do**
　候補の作成
　$\mathcal{N} \leftarrow \{[y, y_{i+1}] \mid y \in \pi[i], y_{i+1} \in V\}$

　スコアの高いもの K 個に絞る
　$\pi[i+1] \leftarrow K - arg\ max_{y \in \mathcal{N}}\ g(y_{i+1}, y_c,\ x) + s(y, x)$
end for
return $\pi[N]$

4. 実験・評価

本節では 3 節で提案した手法を実験し，既存手法と比較評価し，その有効性を示す．

4.1. データセット

産経ニュース（http://www.sankei.com/）から 2011 年 10 月 3 日から 2015 年 11 月 28 日の記事とタイトルのセットを 2015 年 10 月 24 日から 2015 年 11 月 28 日にかけて収集し，タイトルをその記事の一文要約として扱い学習を行った．学習には 10 万記事を用い，うち 9 万記事をトレーニング，1 万記事をバリデーションに

当てた.テストセットとして別途 1000 記事を用意した.語彙数は記事，タイトルでそれぞれ，9 万語と 3 万語となった.

4.2. 比較手法

比較手法として，2.3 の文レベルの生成型要約を用いる．2.3 と同様に記事の最初の一文を元文書として扱ったものに加え，記事全体を元文書として扱ったものを比較手法として用いる.

4.3. 実装

実装には Torch[5]フレームワークを用いた．パラメータは 2.3 の論文で用いられたパラメータに近づけるよう表 **2** のように設定した.初期学習率は 1 エポック目での過学習を避けるため，0.01 とした．最大要約長はデータセットのタイトルの最大長を設定した.

表 2: 各ハイパーパラメータの値

パラメータ	値
エンベディングサイズ D	200
隠れ層サイズ H	400
過去の出力単語数 C	5
スムージングウィンドウ幅 Q	2
初期学習率	0.01
学習率減衰率	0.5
ビームサイズ K	30
最大要約長	18

表 **3** に示した実験環境で実験を行い，学習には約 5 日を要した.

表 3: 実験環境

項目	値
CPU	Intel Core i7-5820K
メインメモリサイズ	16GB
GPU	NVIDIA Quadro K5200
GPU メモリサイズ	8GB

4.4. 評価方法

自動要約の一般的な評価手法として，参照要約とシステム要約の単語の一致で評価する ROUGE[7]が用いられる． ROUGE は単語の一致の評価方法でいくつか種類がある．表 **4** に ROUGE の種類をまとめる.

[5] Torch, http://torch.ch, (2016 年 1 月 5 日アクセス)

表 4: ROUGE の種類

ROUGE の種類	一致の評価方法
ROUGE-N	n-gram の一致で評価
ROUGE-L	最長共通部分列（Longest Common Subsequence）の一致で評価
ROUGE-S	skip-bigram の一致で評価
ROUGE-SU	skip-bigram と uni-gram の一致で評価

ROUGE-N の n-gram のサイズを 1, 2 にした ROUGE-1, ROUGE-2 と ROUGE-L, ROUGE-S, ROUGE-SU で評価を行う． ROUGE は機械的に評価できるが，必ずしも ROUGE が高い場合に良い要約とは言えない．そのため，ROUGE に加え，人手による嗜好テストで評価を行う．人手の嗜好テストは次ように行った.

① 被験者に元記事を読んでもらい，参照要約およびシステム要約をランダムに並び替え，アルゴリズムを伏せた状態で提示する.
② 被験者に提示された要約を元記事の要約としてあっていると感じる順に並び変えてもらう.
③ 被験者につけてもらった順位を元に各要約の優位差を統計的に測定する.

4.5. 実験結果

ROUGE による評価を以下に示す．既存手法の最初の一文を元文書としたモデルを ABS(first line), 元文書全体としたモデルを ABS(article)とし, 我々の提案手法である平均マルチアテンションモデルを Multi-Attention(average), ディープマルチアテンションモデルを Multi-Attention(deep)とする.

表 5: ROUGE による評価.

Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-S	ROUGE-SU
ABS(first sentence)	0.376	0.091	0.246	0.149	0.177
ABS(article)	0.381	0.105	0.253	0.154	0.182
Multi-Attention(average)	0.346	0.080	0.232	0.133	0.160
Multi-Attention(deep)	0.359	0.102	0.245	0.142	0.169
Human Summary	**0.414**	**0.159**	**0.289**	**0.180**	**0.208**

嗜好テストは 4 名の被験者に各 10 件の記事に対する要約を見せ，行った．平均の順位とウェルチ法の T 検定の p 値を表 **6**，表 **7** に示す.

表 6: 平均順位

Model	平均順位
参照要約	1.075
ABS(first line)	3.575
ABS(article)	3.425
Multi-Attention(average)	3.950
Multi-Attention(deep)	2.975

表 7: 提案手法とのウェルチ法の T 検定の p 値

Model	Multi-Attention(deep)
参照要約	8.362×10^{-14}
ABS(first line)	0.017
ABS(article)	0.081
Multi-Attention(average)	0.001

4.6. まとめ・考察

ROUGE の評価では既存手法と提案手法との間に有意差は見られなかったが，嗜好テストによる評価では提案手法であるディープマルチアテンションモデルが既存手法より 90%信頼区間で好ましい要約であると示された．ROUGE は参照要約に対する単語の一致を測るもので，言い換えや並び替えを行う生成型要約では適した評価手法ではなく，そのため，ROUGE の評価では有意差は得られなかったと考えられる．

5. まとめ

本稿では複数のアテンションモデルを用いることにより複数文から一文要約を生成型要約で行う方法を提案した．嗜好テストによる評価を行い，提案手法であるディープマルチアテンションモデルが既存手法より 90%信頼区間で好ましい要約を生成できることを示すことができた．

参 考 文 献

[1] Jing H., “Using Hidden Markov Modeling to Decompose Human-Written Summaries”, Computational linguistic, 2002.

[2] Cho K., Van Merrienboer B., Gulcehre C, Bahdanau D., Bougares F, Schwenk H and Bengio Y., “Learning Phrase Representations using RNN Encoder-Decoder for Statistica Machine Translation”, Conference on Empirical Methods in Natural Language Processing, 2014.

[3] Sutskever Ilya, Oriol Vinyal, and Quoc VV Le, “Sequence to sequence learning with neural networks”, Advances in neural information processing system, 2014.

[4] Bahdanau D., Cho K., Gulcehre C and Bengio Y., “Neural Machine Translation by Jointly Learning to Align and Translate”, Conference on International Conference on Learning Representation, 2015.

[5] M. Rush A., Chopra S. and Weston J., “A Neural Attention Model for Abstractive Sentence Summarization”, Conference on Empirical Methods in Natural Language Processing, 2015.

[6] Mikolov T., Karafiat M., Burget L., Cernocky J. and Kludanpur S., “Recurrent neural network based language model”, Conference of International Speech Communication Association, 2010.

[7] Lin C. W., “ROUGE: A Package for Automatic Evaluation of Summaries”, Proceeding of the ACL-04 workshop, Vol.8, 2004.

[8] Bengio Y., Ducharme R., Vincent P. and Jauvin C., “A Neural Probabilistic Language Model”, The Journal of Machine Learning Research, pp.1137-1155, 2003.